Aufgabenblatt zur Heimarbeit 3

Seminar: Methoden der sozialen Netzwerkanalyse

Mirco Bazzani, Luca Keiser & Amir Shehadeh

Die folgenden Visualisierungen basieren auf den Ständeratsnetzwerken, welche in ersten beiden Heimarbeiten erstellt wurden (siehe Keiser, Shehadeh & Bazzani Heimarbeit 1 & 2). Kurz zusammengefasst stellen die Knoten die einzelnen Ständerät:innen. Die Kanten beziehen sich auf die gemeinsamen Mitgliedschaften in den jeweiligen Lobby-Organisationen. Alle Kanten des Graphen gelten dabei als ungerichtet, da wir davon ausgehen, dass sich die Personen jeweils gegenseitig kennen und wahrnehmen. Dieser Beziehung wird keine positive oder negative Konnotation zugewiesen. In der letzten Arbeit haben wir und dafür entschieden, lediglich jene Knoten zu visualisieren, welche einen Betweenness-Score von mindestens fünf aufwiesen. Diese haben wir für die Visualisierungen der jetzigen Arbeit wieder entfernt. Dafür stellen wir nun Verbindungen zwischen Ständeräten dar, die nicht in der gleichen Partei tätig sind und vergleichen diese mit dem dichteren Netzwerk mit allen innerparteilicher Edges.

Zuerst wurden die Masszahlen für Grösse, Dichte, Komponenten, Durchmesser und Clustering berechnet. Beim Netzwerk handelt es sich jeweils um ein Ein-Komponenten-Netzwerk, auch wenn die innerparteilichen Edges wegfallen. Somit sehen wir, dass die Parlamentarier:innen überparteilich durch die Lobbyorganisationen miteinander verbunden sind. Die Dichte des Netzwerkes mit oder ohne innerparteilichen Edges ist bei einem Score von 0.13 respektive 0.10 (Ratiorange 0 - 1) eher gering. Je näher zur 1 desto grösser wäre das Netzwerk miteinander verbunden (Luke 2015: 26). Der Durchmesser des Netzwerkes, also der größte Abstand zwischen zwei Scheitelpunkten beträgt ohne innerparteilichen Verbindungen 6. Die Grösse beläuft sich mit innerparteilichen Edges auf 118 und ohne innerparteiliche Verbindungen auf 88 (siehe Anhang Masszahlen 1).

Anschliessend wurden die Cutpoints mit dem statnet-Paket identifiziert. Cutpoints bezeichnen Knoten im Netzwerk, welche die Eigenschaft besitzen, dass sich die Komponentenzahl bei deren Wegfallen vergrössert. Im reduzierten Netzwerk der Ständerät:innen wurden insgesamt 9 Cutpoints gefunden (siehe Anhang Cutpoints).  
Brücken die Verbindungslinien zwischen Subgruppen und stellen somit das Edge-Äquivalent zu den Cutpoints dar: Wenn diese Linie bzw. Brücke verschwindet, entstehen mindestens zwei einzelne Komponenten (Luke 2015: 102). Für das reduzierte Netzwerk der Stände:rätinnen konnte der Algorithmus von Luke (siehe Luke 2015: 102f.) insgesamt 8 Bridges identifizieren (siehe Anhang Bridges).

Der nächste Schritt war die Berechnung der Degrees, Closness und Betweenes (siehe Anhang Masszahlen 2). Wie bereits in der Heimarbeit 2 thematisiert (Keiser, Shehadeh & Bazzani, Heimarbeit 2). Dabei entschieden wir uns für die Färbung der Knotenpunkte, um die Parteizugehörigkeit der Parlamentarier: innen zu visualisieren. Darüber hinaus wird die Betweenness eines Knotens durch dessen Durchmesser dargestellt. Zusätzlich wird die Linienstärke der Kanten durch die standardisierte Gewichtung indiziert (Luke 2015: 68) : je dicker die Kanten, desto stärker ist die angenommene Beziehung durch Lobbyzugehörigkeit und Interessensbindungen zwischen den Ständerät:innen. Anhand der Darstellungen lassen sich “zentrale” Akteure innerhalb des Netzwerks identifizieren. Beispielsweise dürfte der hohe Betweenness-Score von Frau Brigitte Häberli-Koller einen Einfluss auf ihre Ressourcenausstattung besitzen (siehe Anhang Visualisierungen). Dabei gilt es zu beachten, dass die Betweenness lediglich eines von mehreren Zentralitätsmassen ist. Hier könnten alternative Zentralitätsmasse (Degree oder Closeness) als Referenz herangezogen werden.

Eine zentrale Aussage dieser Struktur des Netzwerkes ist, dass Vernetzungen zwischen einzelnen Parlamentarier:innen überparteilich über die Lobbygruppen entstehen können. Dies ist sicherlich dem Mehrparteiensystem der Schweiz geschuldet, wo die Interessen nicht klar auf zwei Parteien aufgeteilt werden können. Betrachten wir zwei eher unübliche Verbindungen (zum Beispiel zwischen Ständerat und Germann Hannes (SVP) und Daniel Jositsch (SP)) sehen wir, dass die Beziehungen zwischen Parlamenarier:innen auch aus überparteilichen Interessenverbindungen bestehen. Ergänzend gilt, dass Ständerät:innen eine spezielle Stellung innerhalb des Schweizer Systems inne haben. Ständerät:innen vertreten eher das Interesse der Kantone und nicht der Parteien, denn Sie werden nicht durch das Proporz-, sondern durch das Majorzwahl gewählt. Beim Netzwerk der Ständerät:innen kann man herauslesen, dass es sich um Interessen- und nicht um Parteienpolitik handelt.

**Literatur**

Luke, Douglas A. 2015. A User’s Guide to Network Analysis in R. New York: Spring

Scott, John. 2017. Social Network Analysis. Fourth Edition. London: SAGE.

*Anzahl Wörter - 595*

## Anhang - R-Code

library(tidyverse)  
library(tidygraph)  
library(igraph)  
library(ggraph)  
library(janitor)  
library(data.table)  
library(knitr)  
library(ggforce)  
  
# Einlesen des Datensatzes  
doc <- read\_delim(here::here("Data", "Lobbywatch", "cartesian\_minimal\_parlamentarier\_interessenbindung.csv"))  
  
# Wir interessieren uns nur die Ständerät:innen  
# Zudem möchten wir lediglich ihre ausserparteilichen Verbindungen  
doc\_SR <- doc %>%   
 filter(parlamentarier\_rat == "SR") %>%   
 filter(ardenation\_rechtsform != "Parlamentarische Gruppe")  
   
  
parlamentarier <- doc\_SR %>%   
 select(parlamentarier\_id,   
 parlamentarier\_name,   
 parlamentarier\_partei) %>%   
 mutate(parlamentarier\_partei = case\_when(  
 is.na(parlamentarier\_partei) ~ "Parteilos",  
 TRUE ~ as.character(parlamentarier\_partei))) %>%   
 group\_by(parlamentarier\_name) %>%   
 distinct(parlamentarier\_id,   
 .keep\_all = TRUE) %>%   
 arrange(parlamentarier\_id) %>%   
 rename("id" = parlamentarier\_id) %>%   
 rename("name" = parlamentarier\_name)

# Erstellen der Gruppen  
groupings <- doc\_SR %>%   
 select(parlamentarier\_id,  
 ardenation\_id) %>%   
 mutate(parlamentarier\_id = str\_c(parlamentarier\_id, "",  
 sep = "\_")) %>% # Wird später gebraucht, um die einzelnen IDs wieder voneinander trennen zu können.  
 group\_by(ardenation\_id, parlamentarier\_id) %>%   
 distinct() %>%   
 summarise(sum = n())  
  
setDT(groupings)  
  
# N = Anahl geteilter Organisationsmitgliedschaften. In diesem Schritt erstellen wir das Kantenattribut.  
ties\_SR <- groupings[groupings, on = "ardenation\_id", allow.cartesian = TRUE][parlamentarier\_id<i.parlamentarier\_id, .N, .(pair = paste0(parlamentarier\_id, i.parlamentarier\_id))]

# Erstellen eines tidy-Datensatzes  
ties\_SR <- ties\_SR %>%   
 separate(pair,   
 into = c("from", "to"),   
 sep = "\_") %>%   
 rename("weight" = N) %>%   
 arrange(desc(weight))  
 #filter(weight >= 5) # Es arden nur Verbindungen zwischen Parlamentarier:innen verwendet, die in >= 5 gleichen Organisationen einsitzen.   
  
# Wir wollen keine Intraparty-Edges, deshalb arden diese gelöscht.  
# Intraparty bedeutet: Personen in derselben Partei arden nicht als Edge dargestellt.  
non\_interparty <- c()  
  
# Der Loop schaut, ob zwei Nodes mit einer Verbindung derselben Partei angehören. Falls ja, dann wird der Index  
# des Paares einer Liste hinzugefügt. Diese Liste wird dann aus unserem Datensatz entfernt.  
  
  
for(I in 1:nrow(ties\_SR)){  
   
 from <- ties\_SR$from[i]  
 to <- ties\_SR$to[i]  
 if(parlamentarier[parlamentarier$id == from,3] == parlamentarier[parlamentarier$id == to,3]){  
 non\_interparty <- append(non\_interparty,i)  
 }  
}  
  
ties\_SR\_reduced <- ties\_SR[-non\_interparty,]  
  
  
  
# erstellen eines Tibbles mit angepassten Datentypen  
ties\_SR <- ties\_SR %>%   
 as\_tibble() %>%   
 mutate(from = as.numeric(from),  
 to = as.numeric(to))  
  
  
ties\_SR\_reduced <- ties\_SR\_reduced %>%   
 as\_tibble() %>%   
 mutate(from = as.numeric(from),  
 to = as.numeric(to))  
  
# Knoten  
parlamentarier

## # A tibble: 45 × 3  
## # Groups: name [45]  
## id name parlamentarier\_partei  
## <dbl> <chr> <chr>   
## 1 4 Carobbio Guscetti, Marina SP   
## 2 34 Kuprecht, Alex SVP   
## 3 36 Rechsteiner, Paul SP   
## 4 38 Stöckli, Hans SP   
## 5 66 Caroni, Andrea FDP   
## 6 76 Fässler, Daniel M   
## 7 86 Français, Olivier FDP   
## 8 102 Graf, Maya Grüne   
## 9 125 Jositsch, Daniel SP   
## 10 131 Knecht, Hansjörg SVP   
## # … with 35 more rows

# Kanten  
ties\_SR

## # A tibble: 118 × 3  
## from to weight  
## <dbl> <dbl> <int>  
## 1 315 316 5  
## 2 238 295 3  
## 3 228 86 2  
## 4 382 86 2  
## 5 228 34 2  
## 6 266 66 2  
## 7 227 317 2  
## 8 295 66 2  
## 9 228 402 2  
## 10 295 4 2  
## # … with 108 more rows

ties\_SR\_reduced

## # A tibble: 88 × 3  
## from to weight  
## <dbl> <dbl> <int>  
## 1 315 316 5  
## 2 238 295 3  
## 3 228 86 2  
## 4 382 86 2  
## 5 228 34 2  
## 6 227 317 2  
## 7 295 66 2  
## 8 295 4 2  
## 9 284 315 2  
## 10 315 34 2  
## # … with 78 more rows

# erstellen des Netzwerkobjekts (tidygraph)  
net\_SR <- as\_tbl\_graph(ties\_SR,  
 directed = FALSE) %>%   
 activate(nodes) %>%   
 mutate(id = as.numeric(name)) %>%   
 select(-name)  
  
  
net\_SR\_reduced <- as\_tbl\_graph(ties\_SR\_reduced,  
 directed = FALSE) %>%   
 activate(nodes) %>%   
 mutate(id = as.numeric(name)) %>%   
 select(-name)

## Masszahlen 1

# Generelle Netzwerkangaben  
# Component  
comp\_sr <- count\_components(net\_SR)  
comp\_sr\_red <- count\_components(net\_SR\_reduced)  
# Dichte  
dens\_sr <- edge\_density(net\_SR,  
 loops = FALSE)  
dens\_sr\_red <- edge\_density(net\_SR\_reduced,  
 loops = FALSE)  
# Diameter  
dia\_sr <- diameter(net\_SR,  
 directed = FALSE)  
dia\_sr\_red <- diameter(net\_SR\_reduced,  
 directed = FALSE)  
  
# Size  
size\_sr <- gsize(net\_SR)  
size\_sr\_red <- gsize(net\_SR\_reduced)  
  
# Cliques  
cliq\_sr <- count\_max\_cliques(net\_SR)  
cliq\_sr\_red <- count\_max\_cliques(net\_SR\_reduced)  
  
# Erstellen eines zusammenfassenden Dataframes für die Masszahlen  
labels\_df <- c("Components", "Density", "Diameter", "Size", "Cliques")  
  
measurements\_df <- data.frame(Measurement = labels\_df,  
 `With Intraparty Connections` = c(comp\_sr,dens\_sr,dia\_sr,size\_sr,cliq\_sr),  
 `Without Intraparty Connections` = c(comp\_sr\_red,dens\_sr\_red,dia\_sr\_red,size\_sr\_red,cliq\_sr\_red))  
  
# Tidy Print  
kable(measurements\_df, caption = "Masszahlen der Netzwerke mit und ohne innerparteilicher Beziehungen")

Masszahlen der Netzwerke mit und ohne innerparteilicher Beziehungen

| Measurement | With.Intraparty.Connections | Without.Intraparty.Connections |
| --- | --- | --- |
| Components | 1.0000000 | 1.0000000 |
| Density | 0.1306755 | 0.1022067 |
| Diameter | 5.0000000 | 6.0000000 |
| Size | 118.0000000 | 88.0000000 |
| Cliques | 56.0000000 | 52.0000000 |

## Identifizierung von Cutpoints

library(statnet)

## Loading required package: tergm

## Loading required package: ergm

## Loading required package: network

##   
## 'network' 1.17.1 (2021-06-12), part of the Statnet Project  
## \* 'news(package="network")' for changes since last version  
## \* 'citation("network")' for citation information  
## \* 'https://statnet.org' for help, support, and other information

##   
## Attaching package: 'network'

## The following objects are masked from 'package:igraph':  
##   
## %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,  
## get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,  
## is.directed, list.edge.attributes, list.vertex.attributes,  
## set.edge.attribute, set.vertex.attribute

##   
## 'ergm' 4.1.2 (2021-07-26), part of the Statnet Project  
## \* 'news(package="ergm")' for changes since last version  
## \* 'citation("ergm")' for citation information  
## \* 'https://statnet.org' for help, support, and other information

## 'ergm' 4 is a major update that introduces some backwards-incompatible  
## changes. Please type 'news(package="ergm")' for a list of major  
## changes.

## Loading required package: networkDynamic

##   
## 'networkDynamic' 0.11.1 (2022-04-04), part of the Statnet Project  
## \* 'news(package="networkDynamic")' for changes since last version  
## \* 'citation("networkDynamic")' for citation information  
## \* 'https://statnet.org' for help, support, and other information

## Registered S3 method overwritten by 'tergm':  
## method from  
## simulate\_formula.network ergm

##   
## 'tergm' 4.0.2 (2021-07-28), part of the Statnet Project  
## \* 'news(package="tergm")' for changes since last version  
## \* 'citation("tergm")' for citation information  
## \* 'https://statnet.org' for help, support, and other information

##   
## Attaching package: 'tergm'

## The following object is masked from 'package:ergm':  
##   
## snctrl

## Loading required package: ergm.count

##   
## 'ergm.count' 4.0.2 (2021-06-18), part of the Statnet Project  
## \* 'news(package="ergm.count")' for changes since last version  
## \* 'citation("ergm.count")' for citation information  
## \* 'https://statnet.org' for help, support, and other information

## Loading required package: sna

## Loading required package: statnet.common

##   
## Attaching package: 'statnet.common'

## The following object is masked from 'package:ergm':  
##   
## snctrl

## The following objects are masked from 'package:base':  
##   
## attr, order

## sna: Tools for Social Network Analysis  
## Version 2.6 created on 2020-10-5.  
## copyright (c) 2005, Carter T. Butts, University of California-Irvine  
## For citation information, type citation("sna").  
## Type help(package="sna") to get started.

##   
## Attaching package: 'sna'

## The following objects are masked from 'package:igraph':  
##   
## betweenness, bonpow, closeness, components, degree, dyad.census,  
## evcent, hierarchy, is.connected, neighborhood, triad.census

## Loading required package: tsna

##   
## 'statnet' 2019.6 (2019-06-13), part of the Statnet Project  
## \* 'news(package="statnet")' for changes since last version  
## \* 'citation("statnet")' for citation information  
## \* 'https://statnet.org' for help, support, and other information

## unable to reach CRAN

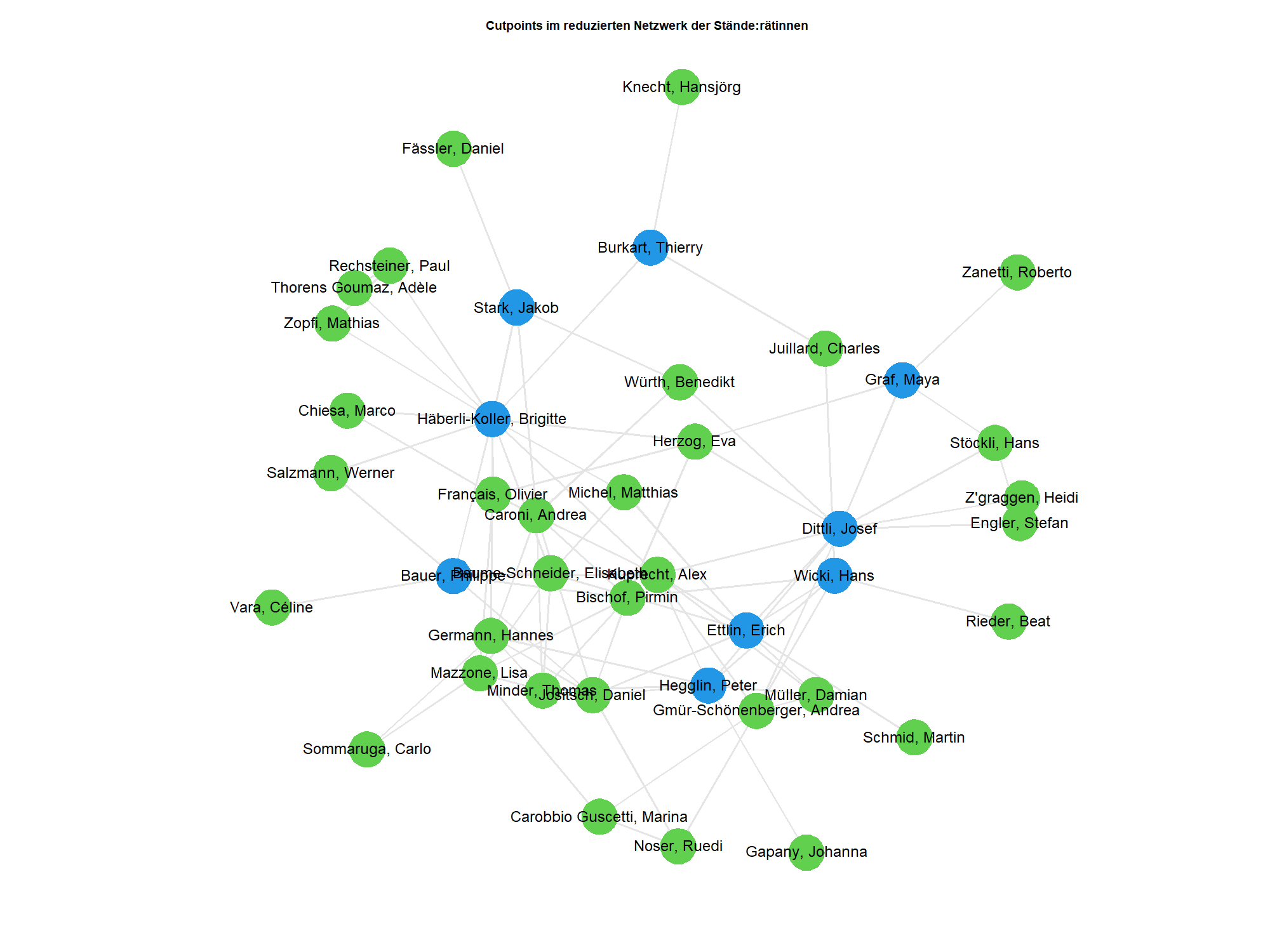
# Workaround  
# erstellen temporärer Objekte zur Zwischenspeicherung  
  
temp\_1 <- ties\_SR\_reduced %>%   
 left\_join(parlamentarier, by = c("from" = "id")) %>%   
 select(name, weight) %>%   
 rename("from" = name)  
  
  
temp\_2 <- ties\_SR\_reduced %>%   
 left\_join(parlamentarier, by = c("to" = "id")) %>%   
 select(name) %>%   
 rename("to" = name)  
  
  
network\_ties\_SR\_reduced <- cbind(temp\_1, temp\_2) %>%   
 relocate(from, to, weight)  
  
  
# löschen der temporären Objekte  
rm(temp\_1, temp\_2)  
gc()

## used (Mb) gc trigger (Mb) max used (Mb)  
## Ncells 2560275 136.8 4597792 245.6 4597792 245.6  
## Vcells 4652074 35.5 10146329 77.5 7614395 58.1

# erstellen eines network-Objekts  
network\_SR\_reduced <- network(network\_ties\_SR\_reduced,   
 type = "edgelist",  
 directed = FALSE)  
  
  
# Identifizierung der Cutpoints  
cutpoints\_SR\_reduced <- cutpoints(network\_SR\_reduced,  
 # graph for undirected graphs  
 mode = "graph",  
 return.indicator = TRUE)  
  
# insegsamt wurden 9 Cutpoints gefunden  
table(cutpoints\_SR\_reduced)

## cutpoints\_SR\_reduced  
## FALSE TRUE   
## 33 9

# Visualisierung  
set.seed(1234)  
gplot(network\_SR\_reduced,  
 gmode = "graph",  
 vertex.col = cutpoints\_SR\_reduced + 3,  
 vertex.border = cutpoints\_SR\_reduced + 3,  
 vertex.cex = 0.7,  
 edge.col = "grey90",  
 jitter = FALSE,  
 displaylabels = TRUE,  
 label.cex = 0.7,  
 label.pos = 5,  
 main = "Cutpoints im reduzierten Netzwerk der Stände:rätinnen",  
 )

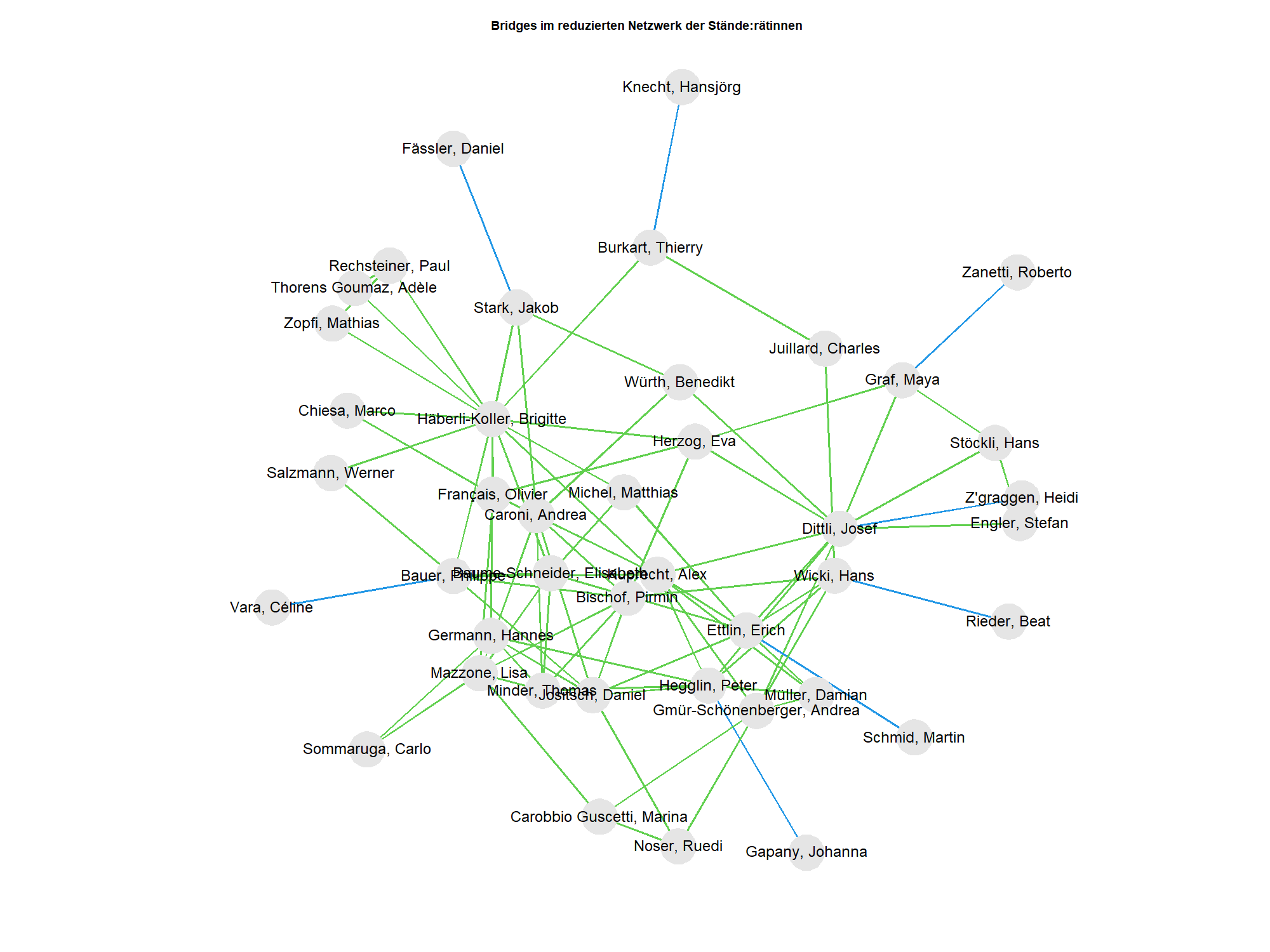


## Bridges

bridges <- function(dat,   
 mode = "graph",  
 connected = c("strong", "weak")) {  
   
 # edgecount  
 e\_cnt <- network.edgecount(dat)  
   
 if(mode == "graph") {  
   
 # get number of components  
 cmp\_cnt <- components(dat)  
   
 b\_vec <- rep(FALSE, e\_cnt)  
   
 for(i in 1:e\_cnt) {  
 dat2 <- dat  
 delete.edges(dat2, i)  
 b\_vec[i] <- (components(dat2) != cmp\_cnt)  
   
 }  
 }  
   
 else {  
 cmp\_cnt <- components(dat,   
 connected = connected)  
   
 b\_vec <- rep(FALSE, e\_cnt)  
 for(i in 1:e\_cnt) {  
 dat2 <- dat  
 delete.edges(dat2, i)  
 b\_vec[i] <- (components(dat2,  
 connected = connected) != cmp\_cnt)  
 }  
 }  
   
 return(b\_vec)  
   
}  
  
network\_SR\_reduced\_bridges <- bridges(network\_SR\_reduced)  
  
# insgesamt wurden 8 Bridges gefunden  
table(network\_SR\_reduced\_bridges)

## network\_SR\_reduced\_bridges  
## FALSE TRUE   
## 80 8

# Visualisierung  
set.seed(1234)  
gplot(network\_SR\_reduced,  
 gmode = "graph",  
 vertex.col = "grey90",  
 vertex.border = "grey90",  
 vertex.cex = 0.7,  
 edge.col = network\_SR\_reduced\_bridges + 3,  
 jitter = FALSE,  
 displaylabels = TRUE,  
 label.cex = 0.7,  
 label.pos = 5,  
 main = "Bridges im reduzierten Netzwerk der Stände:rätinnen",  
 )



## Masszahlen 2

# hinzufügen der Knotenattribute  
net\_SR\_reduced <- net\_SR\_reduced %>%   
 activate(nodes) %>%  
 # Datentyp muss angepasst werden.  
 left\_join(parlamentarier,  
 by = c("id"))  
  
  
# Hinzufügen Zentralitätsmasse  
net\_SR\_reduced <- net\_SR\_reduced %>%   
 activate(nodes) %>%   
 # Berechnung und standardisierung([0,1]) der Betweenness  
 mutate(degree = centrality\_degree(),  
 betweenness = centrality\_betweenness(),  
 closeness = centrality\_closeness())  
  
# Extrahierung der relevanten Zeilen  
masszahlen <- net\_SR\_reduced %>%   
 select(name,  
 parlamentarier\_partei,  
 degree,  
 betweenness,  
 closeness) %>%   
 arrange(desc(degree))  
  
# Tidy Print  
kable(masszahlen)

| name | parlamentarier\_partei | degree | betweenness | closeness |
| --- | --- | --- | --- | --- |
| Häberli-Koller, Brigitte | M | 14 | 274.0845238 | 0.0126582 |
| Dittli, Josef | FDP | 10 | 169.5289683 | 0.0116279 |
| Ettlin, Erich | M | 8 | 84.3039683 | 0.0111111 |
| Hegglin, Peter | M | 8 | 85.9996032 | 0.0111111 |
| Kuprecht, Alex | SVP | 8 | 92.1123016 | 0.0123457 |
| Mazzone, Lisa | Grüne | 7 | 40.5916667 | 0.0101010 |
| Jositsch, Daniel | SP | 7 | 42.8797619 | 0.0108696 |
| Bauer, Philippe | FDP | 7 | 65.8746032 | 0.0111111 |
| Bischof, Pirmin | M | 7 | 47.6178571 | 0.0109890 |
| Minder, Thomas | Parteilos | 6 | 12.5003968 | 0.0103093 |
| Gmür-Schönenberger, Andrea | M | 6 | 45.2928571 | 0.0103093 |
| Wicki, Hans | FDP | 6 | 75.8642857 | 0.0100000 |
| Caroni, Andrea | FDP | 6 | 29.9646825 | 0.0101010 |
| Baume-Schneider, Elisabeth | SP | 6 | 32.2523810 | 0.0111111 |
| Herzog, Eva | SP | 5 | 65.5273810 | 0.0113636 |
| Germann, Hannes | SVP | 5 | 41.9400794 | 0.0108696 |
| Français, Olivier | FDP | 5 | 22.4059524 | 0.0107527 |
| Graf, Maya | Grüne | 4 | 44.1388889 | 0.0090090 |
| Stark, Jakob | SVP | 4 | 57.1583333 | 0.0099010 |
| Müller, Damian | FDP | 4 | 3.3928571 | 0.0096154 |
| Michel, Matthias | FDP | 3 | 6.3317460 | 0.0100000 |
| Rechsteiner, Paul | SP | 3 | 0.5000000 | 0.0085470 |
| Noser, Ruedi | FDP | 3 | 4.0595238 | 0.0086957 |
| Burkart, Thierry | FDP | 3 | 54.2150794 | 0.0090090 |
| Würth, Benedikt | M | 3 | 18.5146825 | 0.0094340 |
| Carobbio Guscetti, Marina | SP | 3 | 6.0825397 | 0.0084746 |
| Stöckli, Hans | SP | 3 | 1.0000000 | 0.0081301 |
| Engler, Stefan | M | 2 | 0.0000000 | 0.0080000 |
| Thorens Goumaz, Adèle | Grüne | 2 | 0.0000000 | 0.0084746 |
| Sommaruga, Carlo | SP | 2 | 0.8333333 | 0.0081967 |
| Chiesa, Marco | SVP | 2 | 0.0000000 | 0.0086207 |
| Zopfi, Mathias | Grüne | 2 | 0.0000000 | 0.0084746 |
| Salzmann, Werner | SVP | 2 | 0.0000000 | 0.0089286 |
| Juillard, Charles | M | 2 | 10.0317460 | 0.0081967 |
| Knecht, Hansjörg | SVP | 1 | 0.0000000 | 0.0066225 |
| Schmid, Martin | FDP | 1 | 0.0000000 | 0.0076923 |
| Rieder, Beat | M | 1 | 0.0000000 | 0.0071429 |
| Zanetti, Roberto | SP | 1 | 0.0000000 | 0.0066225 |
| Gapany, Johanna | FDP | 1 | 0.0000000 | 0.0076923 |
| Fässler, Daniel | M | 1 | 0.0000000 | 0.0070922 |
| Vara, Céline | Grüne | 1 | 0.0000000 | 0.0076923 |
| Z’graggen, Heidi | M | 1 | 0.0000000 | 0.0079365 |

## Cluster

# Erstellen mehrerer Clusterings anhand dreier Algorithmen:  
# Edge Betweenness, Fast Greedy & Infomap  
  
cluster <- net\_SR\_reduced %>%   
 activate(nodes) %>%   
 mutate(group\_edge\_betweenness = group\_edge\_betweenness(),  
 group\_fast\_greedy = group\_fast\_greedy(),  
 group\_infomap = group\_infomap()) %>%   
 select(name,  
 parlamentarier\_partei,  
 group\_edge\_betweenness,  
 group\_fast\_greedy,  
 group\_infomap) %>%   
 arrange(group\_edge\_betweenness)

## Warning in cluster\_edge\_betweenness(graph = .G(), weights = weights, directed =  
## directed): At core/community/edge\_betweenness.c:484 : Membership vector will be  
## selected based on the lowest modularity score.

## Warning in cluster\_edge\_betweenness(graph = .G(), weights = weights, directed  
## = directed): At core/community/edge\_betweenness.c:489 : Modularity calculation  
## with weighted edge betweenness community detection might not make sense --  
## modularity treats edge weights as similarities while edge betwenness treats them  
## as distances.

# Tidy Print  
kable(cluster)

| name | parlamentarier\_partei | group\_edge\_betweenness | group\_fast\_greedy | group\_infomap |
| --- | --- | --- | --- | --- |
| Gmür-Schönenberger, Andrea | M | 1 | 1 | 5 |
| Ettlin, Erich | M | 1 | 1 | 1 |
| Michel, Matthias | FDP | 1 | 2 | 1 |
| Hegglin, Peter | M | 1 | 3 | 2 |
| Noser, Ruedi | FDP | 1 | 3 | 5 |
| Schmid, Martin | FDP | 1 | 1 | 1 |
| Müller, Damian | FDP | 1 | 1 | 1 |
| Kuprecht, Alex | SVP | 1 | 1 | 1 |
| Wicki, Hans | FDP | 1 | 1 | 4 |
| Rieder, Beat | M | 1 | 1 | 4 |
| Carobbio Guscetti, Marina | SP | 1 | 3 | 5 |
| Gapany, Johanna | FDP | 1 | 3 | 2 |
| Minder, Thomas | Parteilos | 2 | 3 | 2 |
| Germann, Hannes | SVP | 2 | 3 | 2 |
| Mazzone, Lisa | Grüne | 2 | 3 | 2 |
| Jositsch, Daniel | SP | 2 | 3 | 2 |
| Bauer, Philippe | FDP | 2 | 2 | 1 |
| Sommaruga, Carlo | SP | 2 | 3 | 2 |
| Bischof, Pirmin | M | 2 | 3 | 2 |
| Caroni, Andrea | FDP | 2 | 3 | 2 |
| Baume-Schneider, Elisabeth | SP | 2 | 2 | 1 |
| Vara, Céline | Grüne | 2 | 2 | 1 |
| Dittli, Josef | FDP | 3 | 1 | 3 |
| Herzog, Eva | SP | 3 | 2 | 3 |
| Engler, Stefan | M | 3 | 1 | 3 |
| Graf, Maya | Grüne | 3 | 1 | 3 |
| Stöckli, Hans | SP | 3 | 1 | 3 |
| Zanetti, Roberto | SP | 3 | 1 | 3 |
| Z’graggen, Heidi | M | 3 | 1 | 3 |
| Häberli-Koller, Brigitte | M | 4 | 2 | 1 |
| Thorens Goumaz, Adèle | Grüne | 4 | 2 | 1 |
| Rechsteiner, Paul | SP | 4 | 2 | 1 |
| Chiesa, Marco | SVP | 4 | 2 | 1 |
| Français, Olivier | FDP | 4 | 2 | 1 |
| Zopfi, Mathias | Grüne | 4 | 2 | 1 |
| Salzmann, Werner | SVP | 4 | 2 | 1 |
| Stark, Jakob | SVP | 5 | 4 | 6 |
| Würth, Benedikt | M | 5 | 4 | 6 |
| Fässler, Daniel | M | 5 | 4 | 6 |
| Knecht, Hansjörg | SVP | 6 | 5 | 4 |
| Burkart, Thierry | FDP | 6 | 5 | 4 |
| Juillard, Charles | M | 6 | 5 | 4 |

## Visualisierung

# Beachten: Um die Cluster darstellen zu können, muss das concaveman-Paket installiert werden  
# install.packages("concaveman")  
  
# Hinzufügen der Kantenattribute  
set.seed(1245)  
  
net\_SR\_reduced %>%   
 activate(nodes) %>%   
 mutate(group\_fast\_greedy = as.factor(group\_fast\_greedy())) %>%   
 ggraph(layout = "fr") +   
 geom\_node\_point(aes(color = parlamentarier\_partei,  
 size = betweenness)) +   
 geom\_node\_text(aes(label = name),  
 repel = TRUE) +  
 geom\_edge\_link(aes(width = weight),  
 show.legend = FALSE) +  
 geom\_mark\_hull(aes(x = x, y = y, fill = group\_fast\_greedy, label = group\_fast\_greedy)) +  
 scale\_size(range = c(5, 15)) +  
 scale\_edge\_width(range = c(0.1, 1)) +   
 scale\_color\_manual(values = c("FDP" = "cornflowerblue",  
 "Grüne" = "chartreuse2",  
 "M" = "darkorange",  
 "SP" = "brown1",  
 "SVP" = "chartreuse4",  
 "Parteilos" = "grey")) +   
 theme\_graph() +   
 labs(title = "Ständerätliches Netzwerk anhand privater Organisationszugehörigkeit - Fruchtermannreingold",  
 subtitle = "Knotenattribute: Namen, Parteizugehörigkeit und Betweenness-Score\nKantenattribute: gemeinsame Einsitze/Mitgliedschaften\nClustering: Fast & Greedy",  
 color = "Partei",  
 caption = "Es werden lediglich Beziehung zwischen Mitgleider:innen unterschiedlicher Parteien visualisiert")

